REMARKS

Claims 1-8 and 19-20 were rejected under 35 USC § 112, second paragraph, as being indefinite. Claims 9-20 were rejected under 35 USC § 101 as being directed to non-statutory subject matter. Claims 1-20 were rejected under 35 USC § 102(b) as being anticipated by Smith et al. (U.S. pat. No. 6,353,906).

Claims 1 and 19 have been amended to remove the word "substantially" to overcome the rejection under 35 USC § 112, second paragraph. Applicant thus requests that the rejection under 35 USC § 112 be withdrawn.

A reporter is being added to claims 9, 19 to claim a tangible result. Basis in the specification for the reporter is found at:

> Delays for generating signals using the functions can be included, such as for randomized delays. Functional simulator 98 keeps track of cycle times and writes the states of various signals to report file 90, or displays signals selected by the designer on a waveform viewer. Signal states could also be compared to expected states and mis-matches reported. (Spec. Para [0107], second half of para.)

Thus claims 9-20 now claims a useful, concrete, and tangible product, a report generated to the user of the simulator. It is well known that such design simulators that report simulation results to users are a multi-million dollar industry and are certainly useful. Applicant thus requests that the rejection under 35 USC § 101 be withdrawn.

PRIOR ART REJECTIONS - 102(B) IN VIEW OF SMITH

Claims 1-20 were rejected under 35 USC § 102(b) as being unpatentable over Smith et al. (U.S. pat. No. 6,353,906).

Smith teaches adding flip-flops and muxes to the design:

> Applying this new methodology to an existing circuit design is straight forward, by simply **substituting an HDL description of the circuit of FIG. 2** wherever a synchronization circuit such as that of **FIG. 1**, appears in the original design. (Smith col. 4, lines 34-38, emphasis added)

Thus Smith teaches that the 5 flip-flops and 1 mux of Fig. 2 be substituted for the 2 flip-flops of Fig 1 at each synchronizer in the design. Three additional flip-flops and one mux are added for each synchronizer.

In contrast to Smith's addition of flip-flops, Applicant does not add flip-flops or muxes. Instead, Applicant adds a single delay element to each synchronizer.

In Applicant's Fig. 5, randomized delay 42 is added between the two existing flip-flops 16, 18. A second synchronizer of flip-flops 26, 28 has randomized delay 44 added between the two existing flip-flops 26, 28.

The added randomized delay 42 simply increases the propagation delay of the preceding flip-flop 16:

> Each of flip-flops 16, 18, 26, 28 has a **clock-to-output propagation delay** that is arbitrarily set to 1 or to some other value. However, first **flip-flop 16 has randomized delay 42 added as an additional delay** before signal SIG_A_SY reaches the D-input of second flip-flop 18. (Spec. Para [0050], emphasis added.)

Randomized delay 42 can be generated by multiplying a random value of 0 or 1 by the clock period T2:

> Randomized delay 42 can be generated by a random function RAND operating on an initial seed S1 to randomly generate either 0 or 1. The result of RAND(S1) is **multiplied** by the **period** of CLK2, T2. The **randomized delay, either 0 or T2, is added** by randomized delay 42 to the delay of first flip-flop 16 so that the **total delay from CLK2 to SIG_A_SY is either 1 or is 1+T2**. (Spec. Para [0050], emphasis added.)

Randomized delay 42 simply increases the total clock-to-out delay of the first flip-flop in the synchronizer to be either 1 or 1+T2. No additional flip-flops or multiplexers need be added to the design, as Smith teaches.

Smith teaches generating the random delay using **flip-flops and multiplexers**. Applicant teaches and claims generating the random delay by **multiplication**:

> wherein the **delay randomizer multiplies** a random binary number by a period of the second clock to **generate** the random delay. (Claim 7)

> wherein inserting an added delay comprises **multiplying** a binary random number with the period of the second clock to generate the added delay. (Claim 15)

Smith nowhere teaches that delays are generated by multiplication. Smith clearly teaches that logic gates such as flip-flops and multiplexers must be added to generate random delays. Thus claims 7, 15 should be allowable.

Claim 4 recites that the delay applicator adds delay parameters:

> wherein the delay applicator adds delay parameters specifying the random delay generated by the delay randomizer to statements in the textual design file for statements that define the first flip-flop of synchronizers of domain-crossing signals.

In contrast, Smith teaches substituting larger synchronizer circuits with additional flip-flops and multiplexers to simulate random delays. Thus Smith cannot anticipate claim 4.

No Random Delay Added to First flip-Flop

Applicant teaches that these random delays are assigned to pre-existing flip-flops in the design file. See Applicant's specification at paragraphs [0074] to [0086]:

> Rather than manually assigning random delays to the synchronizer flip-flops, a simulation tool such as Verilog can be used to randomly assign these delays. The $random function in Verilog can be used with a seed or starting value to initiate the random function.
>
> The following is an example of pseudo-code that randomly assigns delays of either 1 or T2 for the first flip-flop of each synchronizer from CLK1 to CLK2 domains:

```
reg  cycle_delay1;
reg  cycle_delay2;
    ...
always @(posedge resetn)
         begin
         cycle_delay1 = $random(seed1);  // either 0 or 1
         cycle_delay2 = $random(seed2);
         ...
       end

         assign ff_delay1 = (cycle_delay1 * T2) + 1;  // either 1 or T2+1
         assign ff_delay2 = (cycle_delay2 * T2) + 1;
```

In contrast to these relatively easy delay-assignments taught by Applicant, Smith has an example of rather complex Verilog statements for his "X transition logic":

```
module xdata (clk, sig_in, sig_out);
parameter MSB = 0;
 input clk;
 input [MSB:0] sig_in;
```

```
output [MSB:0] sig_out;
reg [MSB:0] sig_out;
reg [MSB:0] prev_sig_in;
always @(sig_in or posedge clk) begin if (sig_in != prev_sig_in) sig_out <= 100'bx;
else sig_out <= sig_in;
prev_sig_in <= sig_in;
end endmodule  (Smith col. 6, lines 20-33)
```

Independent claim 1 recites that the random delay is applied to the first flip-flop:

> a delay applicator, coupled to the delay randomizer, for **applying** the **random delay to a first flip-flop**, the first flip-flop being **clocked by the second clock** but **receiving one of the domain-crossing signals** generated by the first clock as an input;

The claimed "first flip-flop" is clocked by the second clock, but receives a "domain-crossing signal" generated by the first clock. The "domain-crossing signal" corresponds to Smith's "Asynchronous Signal". In Smith's Fig. 2, his "Asynchronous Signal" is only applied to flip-flops 22, 36, so only flip-flops 22, 36 could be the claimed first flip-flop clocked by the second clock.

However, Smith's random delay is not applied to his flip-flops 22, 36. Instead, the random delay is generated by his mux 50, which is several flip-flops 26, 40, 44 downstream from his flip-flops 22, 36. Thus Smith fails to teach the claimed "delay applicator" that applies the random delay to the "first flip-flop". Smith teaches that the added delay is applied several flip-flops downstream, not at the first flip-flop as claimed.

Smith's Fig. 5 Has a Range of Delay Values, Not 2 Discrete Values

In Smith's Fig. 5, his "Asynchronous Signal" is applied to flip-flop 104, so only flip-flop 104 could be the claimed first flip-flop. Mux 108 adds a delay. However, mux 108 adds the wrong delay. Claim 1 recites:

> a delay randomizer that randomly selects as a **random delay either a first delay value or a second delay value**, the first and second delay values ~~substantially~~ **differing** by the **second clock period**;

The claimed delay values differ by the second clock period. Since Smith's mux 108 receives his "Asynchronous Signal" 110 and the synchronized output 106 from flip-flop

104, the delay can have more than two values. Indeed, Smith's delays added by his mux 108 take on a continuous range of values rather than 2 discrete values.

Since Smith's "Asynchronous Signal" 110 and the synchronized output 106 in his Fig. 5 are asynchronous to each other, rather than synchronous as in Smith's Fig. 2, the delay values must necessarily vary over a continuous range, depending on the current phase of his "Asynchronous Signal" 110 to the synchronized output 106.

For example, when 110 arrives at flip-flop 104 just before clock 100 rises, the new state of Asynchronous Signal 110 immediately gets clocked into flip-flop 104, and synchronized output 106 changes very quickly, without waiting for a whole clock period. The delay between Asynchronous Signal 110 and synchronized output 106 is much less than one period of clock 100. The delay between Asynchronous Signal 110 and synchronized output 106 can be as little as the clock-to-out delay of flip-flop 104.

However, when 110 arrives at flip-flop 104 just after clock 100 rises, the new state of Asynchronous Signal 110 does not get clocked into flip-flop 104. It must wait for a whole clock period until the next rising edge of clock 100 to be clocked into flip-flop 104. Then synchronized output 106 changes. The delay between Asynchronous Signal 110 and synchronized output 106 is one clock period of clock 100.

When 110 arrives at flip-flop 104 just about half a clock period after clock 100 rises, the new state of Asynchronous Signal 110 does not get clocked into flip-flop 104. It must wait for a half clock period until the next rising edge of clock 100 to be clocked into flip-flop 104. Then synchronized output 106 changes. The delay between Asynchronous Signal 110 and synchronized output 106 is one-half of the clock period of clock 100.

When 110 arrives at flip-flop 104 just about one-quarter of a clock period after clock 100 rises, the new state of Asynchronous Signal 110 does not get clocked into flip-flop 104. It must wait for three-quarters of the clock period until the next rising edge of clock 100 to be clocked into flip-flop 104. Then synchronized output 106 changes. The delay between

Asynchronous Signal 110 and synchronized output 106 is 3/4 of the clock period of clock 100.

When 110 arrives at flip-flop 104 just about three-quarters of a clock period after clock 100 rises, the new state of Asynchronous Signal 110 does not get clocked into flip-flop 104. It must wait for one-quarter of the clock period until the next rising edge of clock 100 to be clocked into flip-flop 104. Then synchronized output 106 changes. The delay between Asynchronous Signal 110 and synchronized output 106 is 1/4 of the clock period of clock 100.

These examples show that the difference in delay between Asynchronous Signal 110 and synchronized output 106 varies. The delay difference can be nearly zero (the clock-to-out delay), 1/4 of the clock period, 1/2 the clock period, 3/4 the clock period, or one whole clock period. Indeed, the delay difference can be any value, from 0 to 1 clock period, including 1/8, 4/5, 5/9, 7/8, or any other fraction of the clock period between 0 and 1.

Since the difference between the two delays selected by Smith's mux 108 can vary continuously over the range of 0 to 1 clock period, including all the many intermediate values, Smith teaches away from claim 1's delay randomizer, which claims that the 2 delay values differ by the second clock period:

> a delay randomizer that randomly selects as a random delay either a first delay value or a second delay value, the first and second delay values ~~substantially~~ differing by the second clock period;

Smith's circuit in his Fig. 5 produces 2 delays that can differ by any value in a range, rather than differing by the second clock period. Thus Smith cannot anticipate claim 1.

Claim 6 further defines that these delay values must differ by the second clock period:

> wherein the first delay value is zero and the second delay value is the second clock period, or wherein the first delay value is an arbitrary delay value and the second delay value is the arbitrary delay value added to the second clock period.

Since Smith's Fig. 5 produces delay values that vary all over the map between 0 and 1 clock period, Smith cannot anticipate claim 6.

In view of the above, it is submitted that claims 1-20, as amended, are in a position for allowance. This application was filed with formal drawings that have not been amended. Applicant believes that a full and complete response to the office action has been made. Reconsideration and re-examination is respectfully requested. Allowance of the claims at an early date is solicited.

If the Examiner believes that a telephone interview would expedite prosecution of this application, he is invited to telephone the undersigned at (831) 476-5506.

Respectfully Submitted,

/Stuart T Auvinen/

Stuart T. Auvinen
Agent for Applicant
Reg. No. 36,435

Stuart T. Auvinen
429 26th Avenue
Santa Cruz, CA 95062

(831) 476-5506
(831) 477-0703 Fax